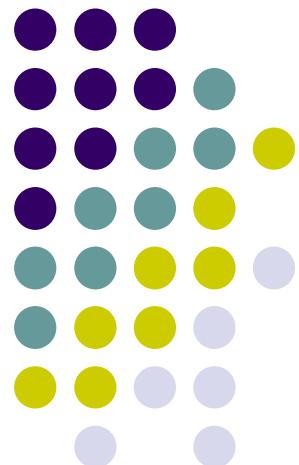


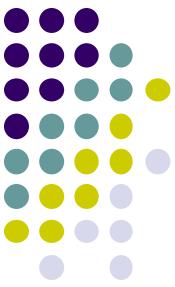
WIX1002

Fundamentals of Programming

Chapter 1

Problem Solving in Programming





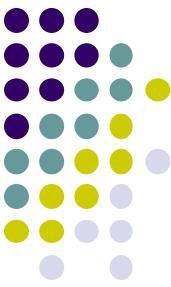
Contents

- Introduction
- Java
- Problem Solving
- Input Process Output
- Pseudocode
- Flow Chart
- Sample Java Program



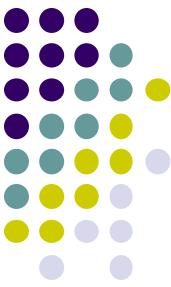
Introduction

- A computer can be define as an electronic machine, operating under the **control of instructions** stored in its own memory, that can accept data, manipulate that data and produce results that can be stored for future use.
- These set of instructions is called **program**.
- A program contains a large number of operations and a computer must be programmed to perform different tasks.
- A programming language can be divided into **machine language, assembly language and high level language**.



Introduction

- **Machine Language**
 - It is the natural language of a particular computer.
 - It is defined by the hardware design of the computer.
 - It consists of strings of numbers (**0 and 1**) that construct computers to perform their operations.
 - It is **machine dependent**, particular machine language can be used only for one type of computer.
 - It is cumbersome for humans.



Introduction

- **Assembly Language**
 - It is the English-like abbreviations languages.
 - Translator programs called **assemblers** were developed to convert assembly language programs to machine language at computer speeds.
 - It is more clearer and more easier to understand as compare to machine language.
 - However, it requires **many statements** in order to perform a simple tasks.
 - Example, **MOV AL, 88h**



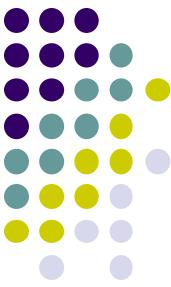
Introduction

- **High Level Language**
 - It is developed to speed the programming process.
 - It was developed in which single statements could be written to accomplish substantial tasks.
 - **Compiler** is used to convert high level language programs into machine language.
 - High level languages allow programmers to write instructions that look almost like everyday English and contain commonly used mathematical notations.
 - Java is one of the High Level language.



Java

- Java is an **Object-Oriented Programming** language.
- In **OOP**, all the things around us is made up of **objects**, such as people, buildings, vehicles, and etc. Each of the object will has the ability to perform certain **actions** and these actions will has some effect on some other objects in the world. Example a people driving a car.
- Java is a well-know programming language for different applications such as console application, desktop application, distributed application, Internet application and mobile application.
- Besides, Java can be used to develop **Java applet** which can be executed in Web browser or applet viewer.



Java

- Java programs normally go through five phases to be executed. These are: **edit, compile, load, verify and execute.**
- **Phase 1 - Edit**
 - This phase consists of editing a file. This is accomplished with an editor program or **Integrated Development Environment (IDE)**. Some of the popular IDE includes **NetBeans, Eclipse, Jcreator** and **JDeveloper**.
 - Java program file names end with **.java** extension. The file name must be same as the **Class name**.



Java

- **Phase 2 – Compile**
 - In this phase, the Java compiler translates the Java program into **byte codes**. The compiler will check whether the source code has the correct spelling and punctuation, all the data types are correct, and all the variables names are legal.
 - Java Platform (**JDK 15**) provides a compiler you can use to compile all kinds of Java programs.
 - Command use to compile java file
 - **javac filename.java**
 - **Syntax** – The **grammar rules** of the language. It tells what arrangement of words and punctuations are allowed in a Class or program definition.



Java

- **Phase 3 – Load**
 - The class loader will take the **.class** file containing the byte codes and transfers it to memory. Java applications are loaded into memory and executed using the Java interpreter.
 - Java interpreter is sometimes referred to as the "Java Virtual Machine" or "Java run-time system".
- **Phase 4 – Verify**
 - In this phase, the byte code verifier is used to verify the byte code before executed. It ensures byte codes do not violate security requirements.



Java

- **Phase 5 – Execute**
 - After verification, the computer will interpret the program one byte code at a time, thus performing the actions specified by the program.
 - Command use to execute java file
 - **java filename**
 - A Java Archive (JAR) file can be created for a Java application that consists of multiple java files.
 - Command use to execute jar file
 - **java –jar filename.jar**



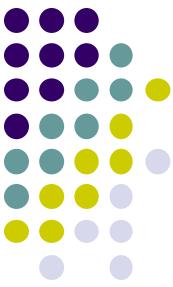
Problem Solving

- The purpose of writing a program is to solve a problem or performing certain tasks.
- Before writing a program to solve a problem, it is essential to have a clear understanding of the problem and a carefully planned approach to solve a problem.
- Any computing problem can be solved by executing a series of actions in a **specific order**.
- A procedure for solving a problem in terms of the actions to be executed and the order in which actions are to be executed is called an **algorithm**.



Problem Solving

- Below are the general steps of problem solving in programming
 - First, understand the problem. Solve the problem manually with a few examples.
 - Then, devise an algorithm to solve the problem.
 - After that, write the program using the programming syntax and compile the program.
 - Correct the syntax error after the compilation. Save the program and compile again.
 - Execute the program if the program is error free.
 - Finally test the results against expected output. If the results are not correct, modify the algorithm, rewrite and recompile the program



Problem Solving

- **Strategies**

- What do I know about the problem?
- What is the information that I have to process in order to find the solution?
- What does the solution look like?
- What sort of special cases exist?
- How will I recognize that I have found the solution?



Problem Solving

- **Syntax** – A set of rules, principles, and processes that govern the **structure of statement** in a programming language.
- **Semantic** – Describe the **meaning** of the things written while following the syntax rules of the language.
Semantic describes the things happen when a program is executed.
- **Debugging** – A process of **eliminating mistakes** in the program. A mistake in a program is called a bug.



Problem Solving

- There are three commonly types of bugs or errors.
- **Syntax Error**
 - A grammatical mistake in the program. A mistake in the arrangement of words and punctuations.
- **Logic Error**
 - A mistake in the underlying algorithm or semantic error.
- **Run-time Error**
 - An error that happen when the program is executed



Input Process Output

- A program usually receives inputs from a user or other source (files), does some computations on the inputs (process), and returns the results of the computations (output).

| Input | Process | Output |
|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <ul style="list-style-type: none">Number 1Number 2Add, Subtract, Multiply or Divide | <ul style="list-style-type: none">Assign Variable for Number 1Assign Variable for Number 2Select Case for Calculation: Add, Subtract, Multiply or DivideCalculate Number 1 and Number 2 | <ul style="list-style-type: none">Result of Calculation |

| Input | Processing | Output |
|------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">Radius of circle | <p>Processing Items:</p> $\text{area} = \pi * \text{radius}^2$ $\text{circumference} = 2 * \pi * \text{radius}$ <p>Algorithm:</p> <p>Step 01: Start Step 02: Input <i>radius</i> from the user Step 03: set <i>PI</i> = 3.1415 Step 04: Calculate area as: <i>area</i> = <i>PI</i> * <i>radius</i>² Step 05: Calculate circumference as: $\text{circumference} = 2 * \text{PI} * \text{radius}$ Step 06: Print <i>area</i> and <i>circumference</i> Step 07: End</p> | <ul style="list-style-type: none">Area of circleCircumference of circle |



Pseudocode

- Pseudocode is an informal high-level description of the operating principle of a computer program or algorithm. It is simply a numbered list of instructions to perform some task.
- Best Practices
 - Write only one statement per line and each statement should express just one action.
 - Get the name of the participant.
 - Compute the total salary.
 - Display the number of students.



Pseudocode

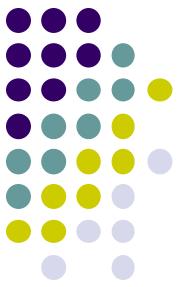
Compute Final Price

1. Get the price of the item
2. Get the sales tax rate
3. Sales Tax = price of item * sales tax rate
4. Final Price = price of item + Sales Tax

- Indent statements that fall inside a selection or loop structure.

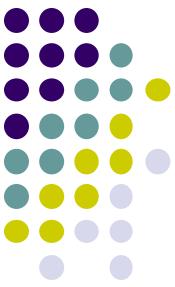
```
If the number of students are more than 40
    Create extra classes
Otherwise
    Create one class
```

```
while the water temperature is greater than 50
    add more ice
    display the new water temperature
```



Flow Chart

- A flowchart is a type of diagram that represents an algorithm or process.
- A flow chart shows the steps as boxes of various kinds, and their order by connecting them with arrows.
- It is used in analysing, designing, documenting or managing a process or program.



Flow Chart



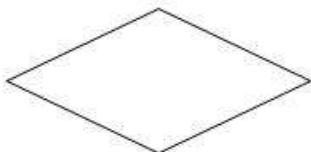
Start/end



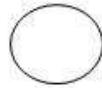
process



Input / output



decision



connector



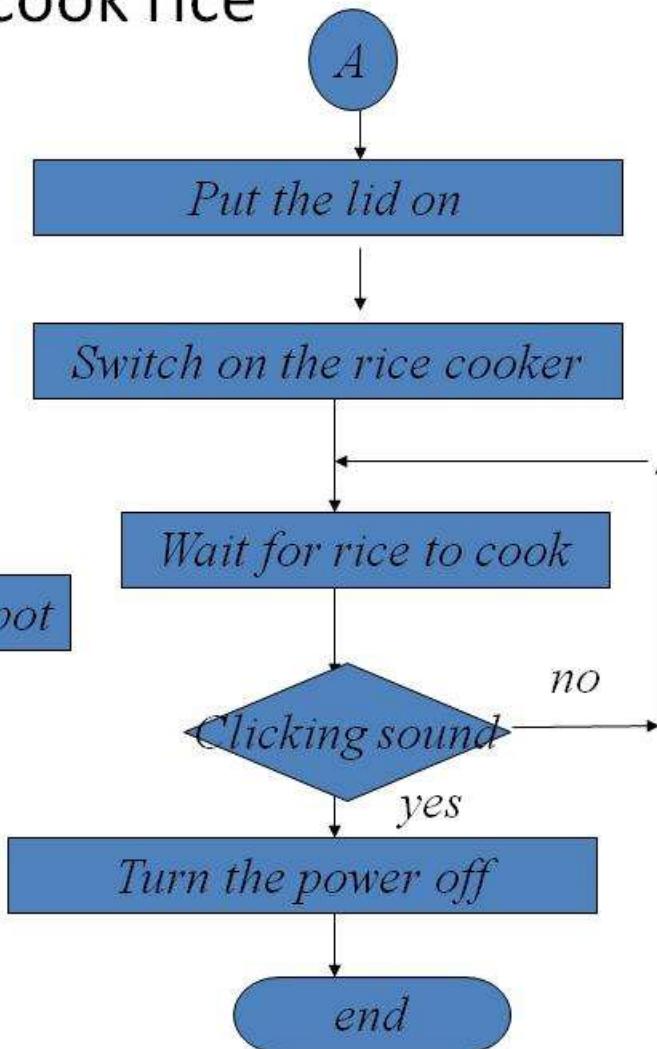
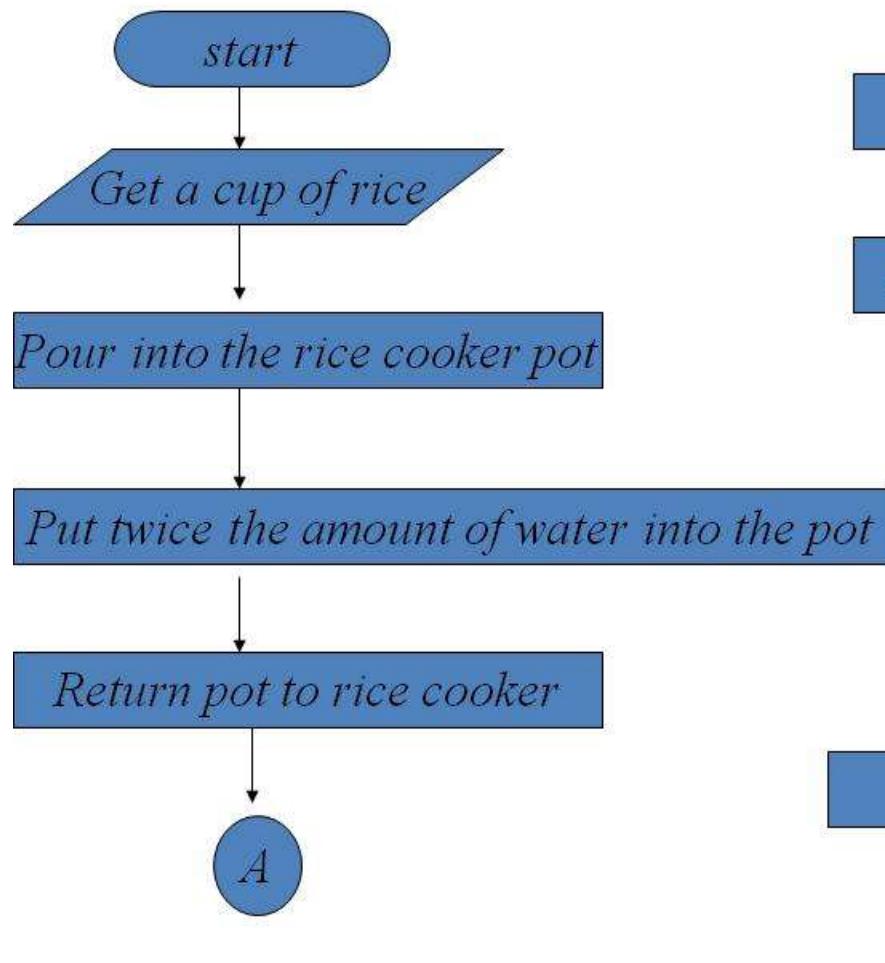
Activity flow

Flow Chart Notation



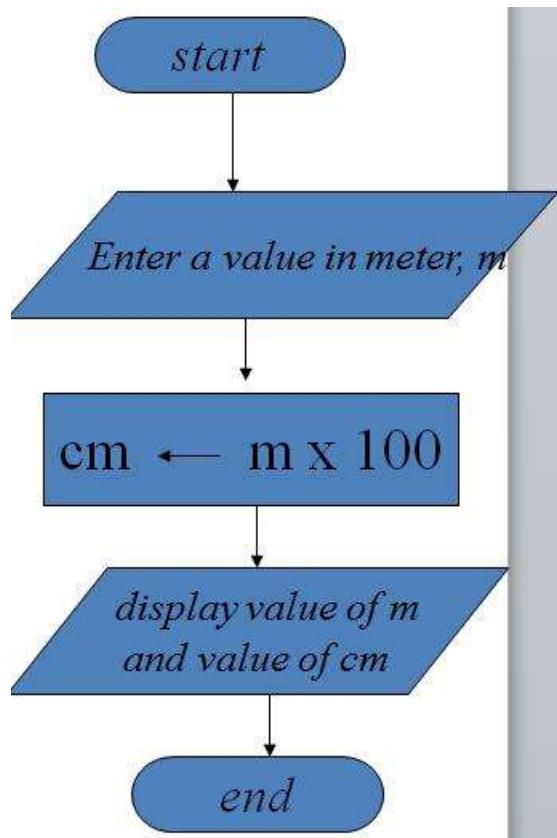
Flow Chart

Flowchart : How to cook rice

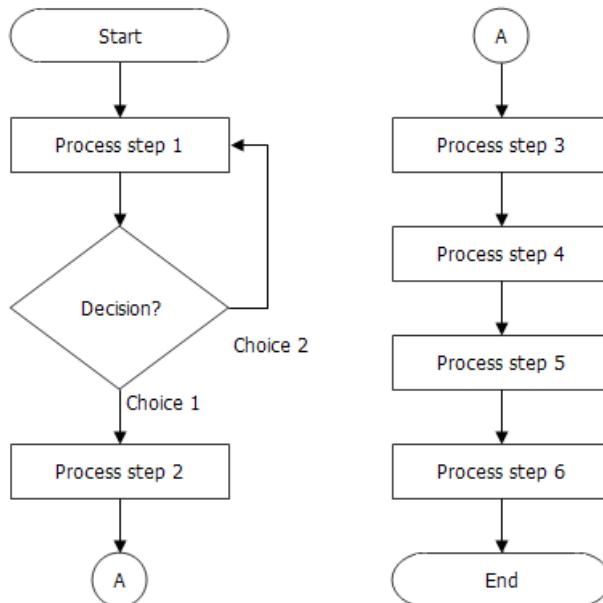




Flow Chart

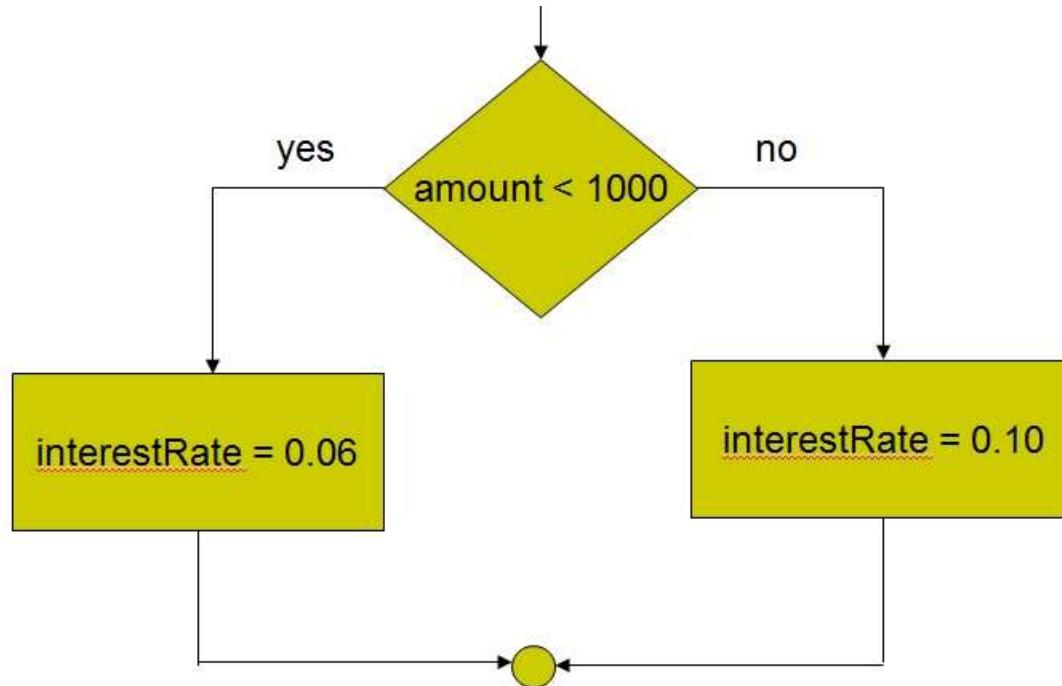


Basic Flowchart

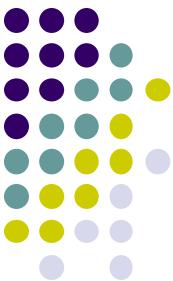




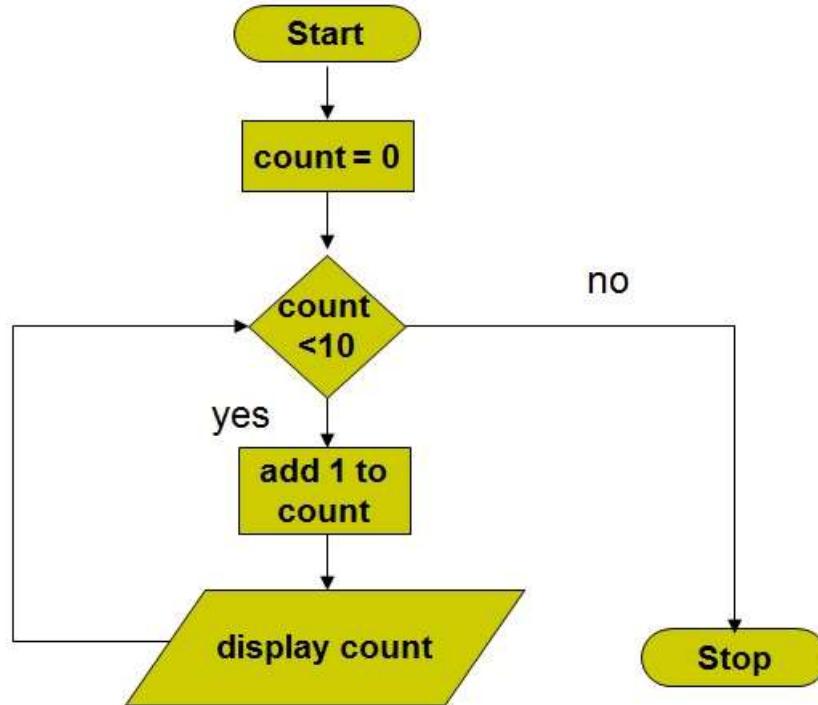
Flow Chart



Selection Structure



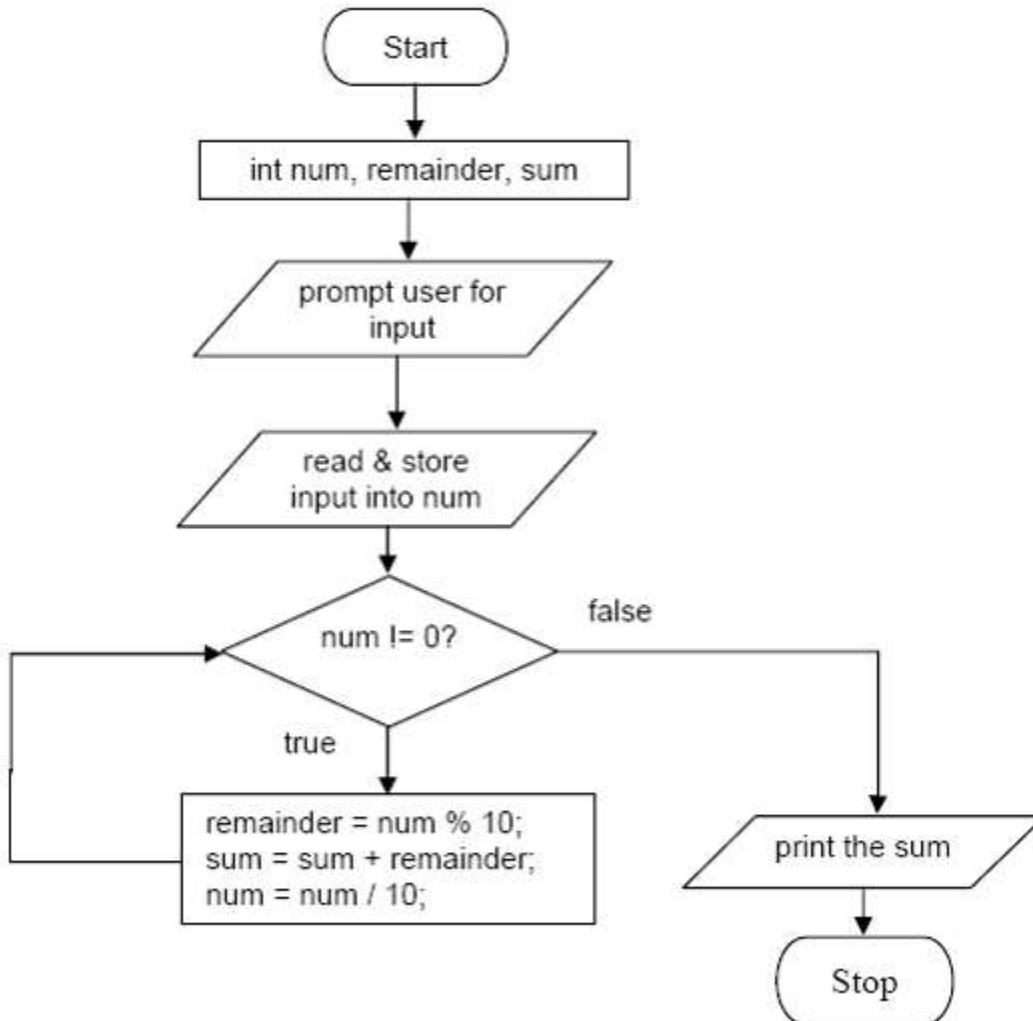
Flow Chart



Repetition Structure



Flow Chart





Sample Java Program

```
public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to Java!");
    }
}
```

* File Name: **FirstProgram.java**



Sample Java Program

- Every program in Java consists of **at least one class definition**. The class name of the program is **FirstProgram**.
- A left brace { - begins the body of every class definition. A right brace } - ends each class definition.
- The class contains a method named **main**. When the program is run, the **main** method is invoked which means the statements in the **main** method are executed.
- The **void** indicates that this method will perform a task and will not return any information when the task is completed.
- The statement in the sample program display the output **Welcome to Java!**



Sample Java Program

- In Java:
 - A program is made up of one or more **classes**
 - A class contains one or more **methods**
 - A method contains program **statements**
- Java is **case sensitive**. The uppercase letters and lowercase letters must be enter correctly.
- Class are the fundamental building blocks of Java program. Each program begin with a **class definition**.
- Every **Java application** contains a **main method**. The instructions in the main method are executed when the application starts. main method must always be **static**.
- Each Java statements must end with a semicolon (;

